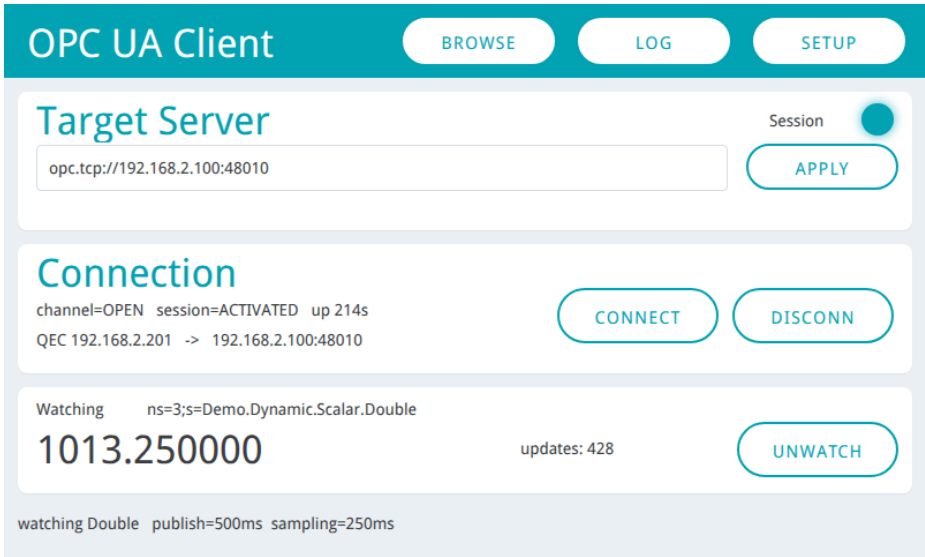


QEC OPC UA Client 測試工具 — 使用手冊 v 1.0

用 86Duino 連上任何一台 OPC UA Server，直接在面板上瀏覽、讀寫、訂閱它的節點

開發指南 | 版本 1.0 | September 2026

ICOP Technology Inc.



封面 Home 頁：session 正在跑，並且訂閱著一顆節點。

修訂紀錄

版本	日期	說明
1.0	September 2026	首版。對應 QEC_OPCTUA_CLIENT_HMI 釋出版：非阻塞連線、面板上的節點瀏覽器與即時值、依型別切換的寫入、最愛。

目錄

修訂紀錄	1
目錄	2
1. 概觀	4
1.1 系統架構	4
1.1.1 硬體需求	4
1.1.2 通訊架構	5
2. 建置 QEC MDevice.....	7
2.1 PC 端安裝	7
2.1.1 86Duino Coding IDE 502	7
2.1.2 open62541 library.....	7
2.1.3 UaCPPServer（用來對測，選配）	7
2.2 程式架構說明	9
2.2.1 網路設定	10
2.2.2 HMI	11
2.2.3 應用邏輯	11
2.2.4 排程與執行	11
2.2.5 OPC UA Client	12
2.3 上傳程式	12
3. 連上 Server 與操作節點	14
3.1 Endpoint.....	14
3.2 瀏覽位址空間	14
3.3 節點屬性	15
3.4 支援寫入的型別	15
3.5 訂閱	15
4. HMI 頁面功能說明.....	17
4.1 Home — 連線.....	17
4.2 Browse — 找節點.....	18
4.3 Node — 用節點.....	20
4.4 Log — 面板上的即時 Console	22
4.5 Setup — QEC 自己的網路與訂閱間隔.....	23
CONFIRM 的驗證規則.....	23
二次確認	24
5. 實際跑一次：連上第一台 Server	26
5.1 這一章要做什麼	26
5.2 開始之前	26
任何一台 OPC UA Server 都能用	26
5.3 步驟	27
5.4 連上之後	27
5.5 動手做四件事	28
過程中會一併看到的現象.....	28
5.6 這些訊息是正常的	28
6. 設定的持久化	30
6.1 什麼要重開機、什麼不用.....	30
6.2 為什麼目標 URL 放在 Home 頁	30
6.3 EEPROM 版本變更	30
7. 疑難排解	31
8. 完整程式與下載	33
9. 後續延伸	34
9.1 對上現場既有的 Server	34
9.2 兩台 QEC 對接.....	34
9.3 QEC 反過來當 Server	34
9.4 可以再延伸的功能	34
9.5 上線之前要處理的安全性.....	35
10. 聯絡我們	36

1. 概觀

QEC_OPCUA_CLIENT_HMI 是一支跑在 **QEC-M** 系列自帶螢幕的主站上的 OPC UA Client 測試工具。燒錄後在面板上輸入目標 endpoint、按 CONNECT，就能走訪對方的位址空間、讀寫數值、訂閱節點——**全程不需要 PC**。

它是 Server 測試工具的鏡像：那邊是 QEC 等 SCADA 連進來，這邊是 QEC 主動連出去。

整套工具分三塊，全部跑在同一顆 **Vortex86EX2** 上：

組成	實作	說明
OPC UA Client	open62541 v1.1.2 (86Duino / DMP 移植版)	非同步連線、Browse、Read、Write、Subscription
HMI 面板	觸控面板、LVGL86 + 86HMI 設計工具	五頁：Home / Browse / Node / Log / Setup
設定持久化	EEPROM bank 0	QEC 自己的網路、目標 URL、6 筆最愛、訂閱間隔

不需要 PC 端的 OPC UA Client、不需要在機台旁邊放一台筆電、也不需要另外一塊 HMI。

這支工具存在的理由有三個：

1. 證明 QEC 能在廠內網路上連到第三方 OPC UA Server
2. 確認那台 Server 提供哪些節點——沒有文件、手邊也沒有筆電時，直接在面板上瀏覽位址空間
3. 在動手寫自己的應用之前，先驗證節點的資料型別、存取權限與訂閱行為

1.1 系統架構

1.1.1 硬體需求

元件	建議規格	說明
QEC MDevice	QEC-M 系列自帶螢幕的主站	本手冊的截圖以 QEC-M-070T (Vortex86EX2、7" 800×480 電阻式觸控) 為例，系列中其他自帶螢幕的型號同樣適用
網路	1× Gigabit RJ45	OPC UA 走這個孔。另外 2× 10/100 RJ45 是 EtherCAT 專用，插錯不會有連線
OPC UA Server	支援 opc.tcp 的都可以	PC 上的 demo server、第三方 PLC，或另一台跑 Server 測試工具的 QEC
電源	依機型規格	現場若無法斷電重開，面板的 REBOOT 鈕可以軟體重開機

軟體需求

元件	版本	說明
86Duino Coding IDE	502	QEC 官網公告版本
open62541 library	v1.1.2	由 IDE 的 Library Manager 安裝 (搜尋 OPC)

元件	版本	說明
86HMI 設計工具	隨 IDE 附帶	產生 <code>myhmi.h / myhmi.cpp</code>
用來對測的 OPC UA Server	—	一般用 Unified Automation 的免費 demo server；本手冊的範例是 <code>opc.tcp://192.168.2.100:48010</code>

換到 QEC-M 系列的其他機型：程式本身不用改。HMI 版面是依 800×480 排的，若你的機型解析度不同，用 86HMI 設計工具重新排一次版即可（見 §8）。另外，開機 banner 裡的 "QEC-M-070T" 是寫死的字串，換機型時可以一併改掉。

1.1.2 通訊架構

OPC UA 的連線永遠是 Client 主動、Server 被動，兩個 Client 之間無法互連。所以「誰是 Server」這件事決定了整個系統的連線架構。



圖 1-1 系統架構 / 資料流圖 (1024×576)。

這支工具裡 QEC 是 Client。分辨兩種角色很簡單，看 sketch 用的是哪一組 API：

```
UA_Client_new() / UA_Client_connectAsync() / UA_Client_Service_read() -> QEC is the client (this manual)
UA_Server_new() / UA_Server_run_startup() / UA_Server_writeValue() -> QEC is the server
(separate manual)
```

兩種角色可以在同一支 sketch 裡共存（對上位系統當 Server、對下游裝置當 Client），library 兩邊都支援。

非阻塞是刻意設計，而且在 Client 這邊比 Server 更關鍵。`UA_Client_connect()` 是阻塞呼叫：對方連不上時，它會一路阻塞到 `cc->timeout` 為止；這段期間 `handleHmiEvents()` 與 `helperYield()` 都不會執行，面板完全沒有反應。這支工具改走非同步路徑：

```
/* ... doConnect(), §8 ... */
UA_StatusCode rc = UA_Client_connectAsync(client, cfgUrl);
...
/* ... loop(), §16 ... */
/* advances the handshake, then carries Publish and DataChange traffic */
UA_StatusCode rc = UA_Client_run_iterate(client, ITER_MS);
```

`UA_Client_connectAsync()` 只做初始化；真正的 TCP 連線發生在 `UA_ClientConnectionTCP_poll()` 裡面，而它的 `select()` timeout 就是 `loop()` 傳給 `UA_Client_run_iterate()` 的那個值。所以不論對方在不在，單一圈最多只阻塞 20 ms。

2. 建置 QEC MDevice

2.1 PC 端安裝

2.1.1 86Duino Coding IDE 502

1. 從 QEC 官網下載並安裝 **86Duino Coding IDE 502**
2. 用 USB 線連接 QEC MDevice 與 PC
3. 開啟 Windows **裝置管理員**，確認「連接埠 (COM port)」底下出現 QEC 的 COM port（範例中是 **COM12**）
4. IDE 的 **Tools** → **Board** 選你手上的 QEC-M 型號（本手冊是 **QEC-M-070T**）；**Tools** → **Port** 選上一步看到的 COM port

2.1.2 open62541 library

1. **Sketch** → **Include Library** → **Manage Libraries...**
2. 搜尋欄輸入 **OPC** → 找到 **open62541** → **Install**
3. 安裝完成後，**Sketch** → **Include Library** 的清單裡會出現 **open62541**，版本 **1.1.2**

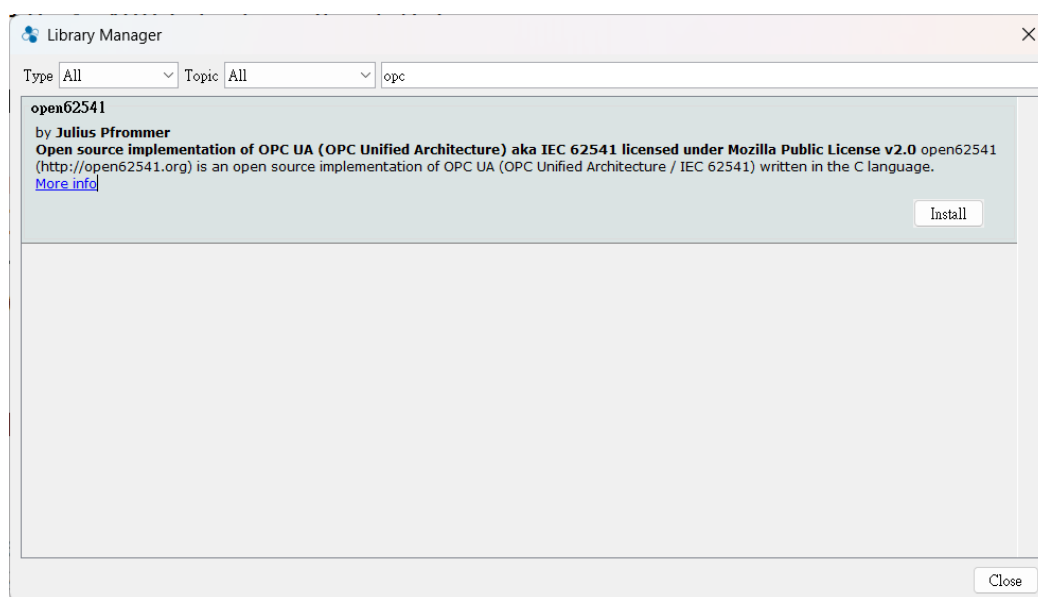


圖 2-1 Library Manager 搜尋opc 之後，清單裡的open62541 就是要裝的library；點Install，裝完在Sketch → Include Library 會看到版本 1.1.2。Server 與 Client 兩支工具用的是同一個library。

離線安裝：把 **open62541** 資料夾直接複製到 IDE 的 **libraries** 底下，重開 IDE。

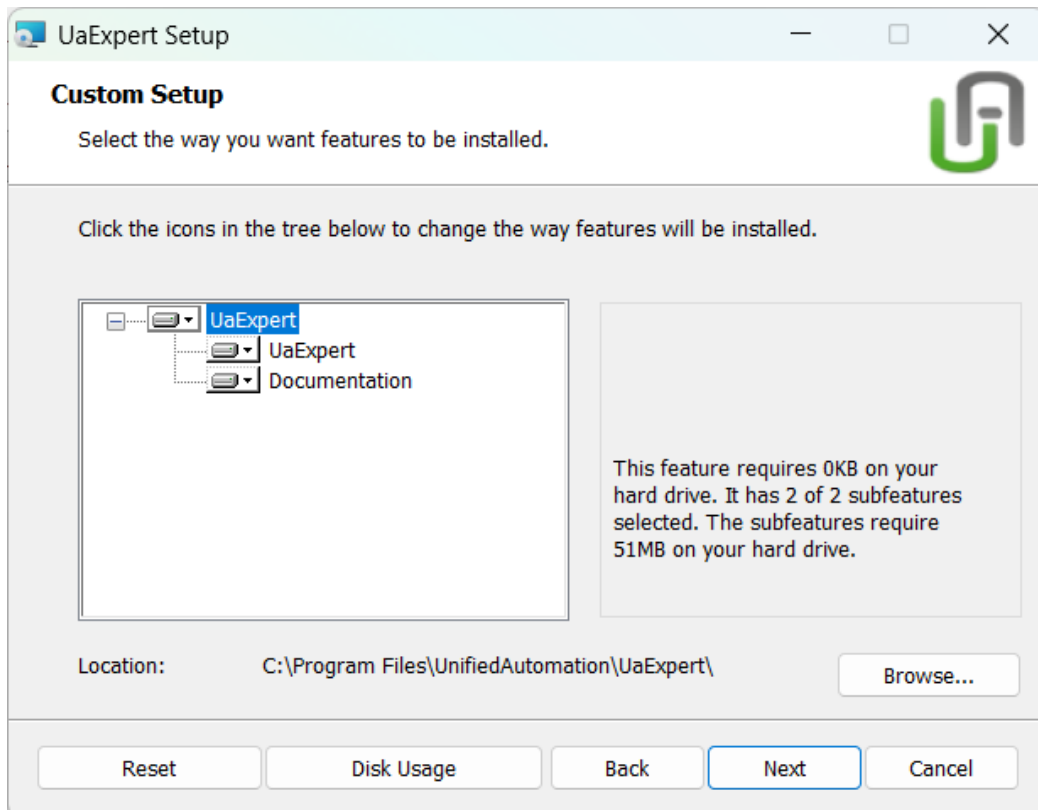
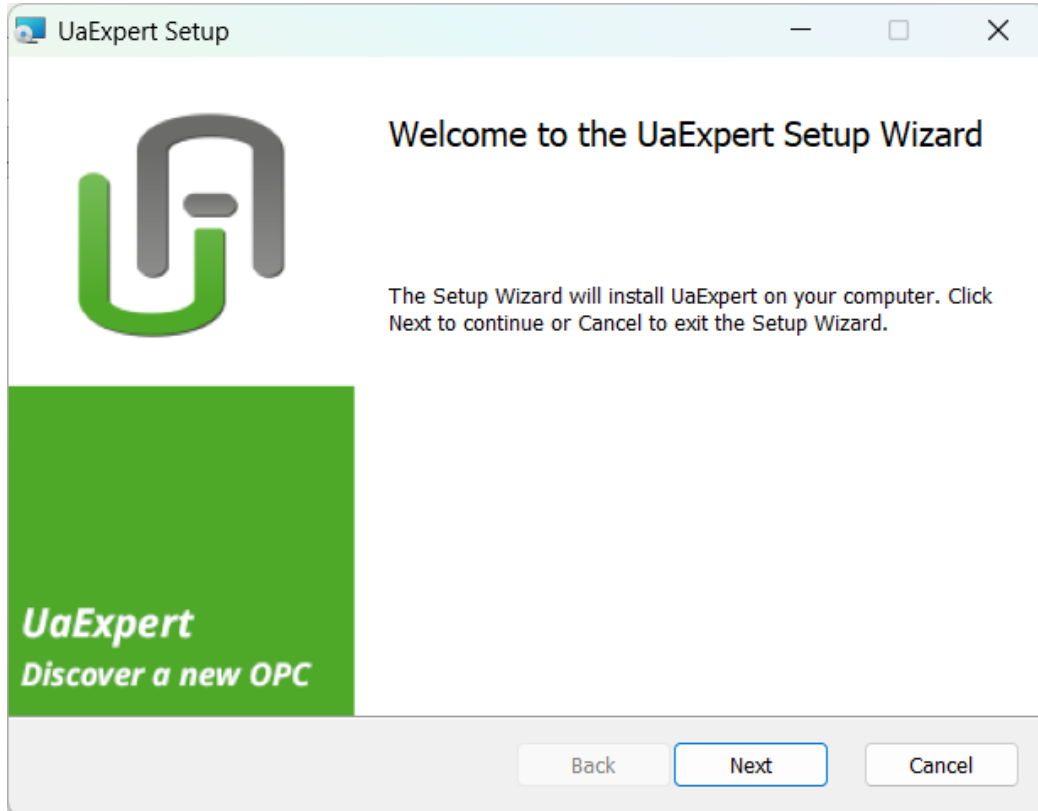
升級注意：機器上若還留著更早的版本（資料夾名為大寫 **Open62541**），請先刪掉。新版 **library.properties** 的 **name** 是小寫 **open62541**，兩份並存時 IDE 可能挑到舊的。

2.1.3 UaCPPServer（用來對測，選配）

這支 Client 需要一台 OPC UA Server 當對象。手上如果還沒有，最快的一條路是 Unified Automation 的 demo

server **UaCPPServer** —— 它隨 UaExpert 的安裝包一起裝上。

1. 到 Unified Automation 網站註冊帳號（免費版就夠）
2. Downloads → OPC UA Clients → UaExpert → 下載並安裝
3. 安裝精靈一路 Next : Custom Setup 保持預設 → Install → Finish



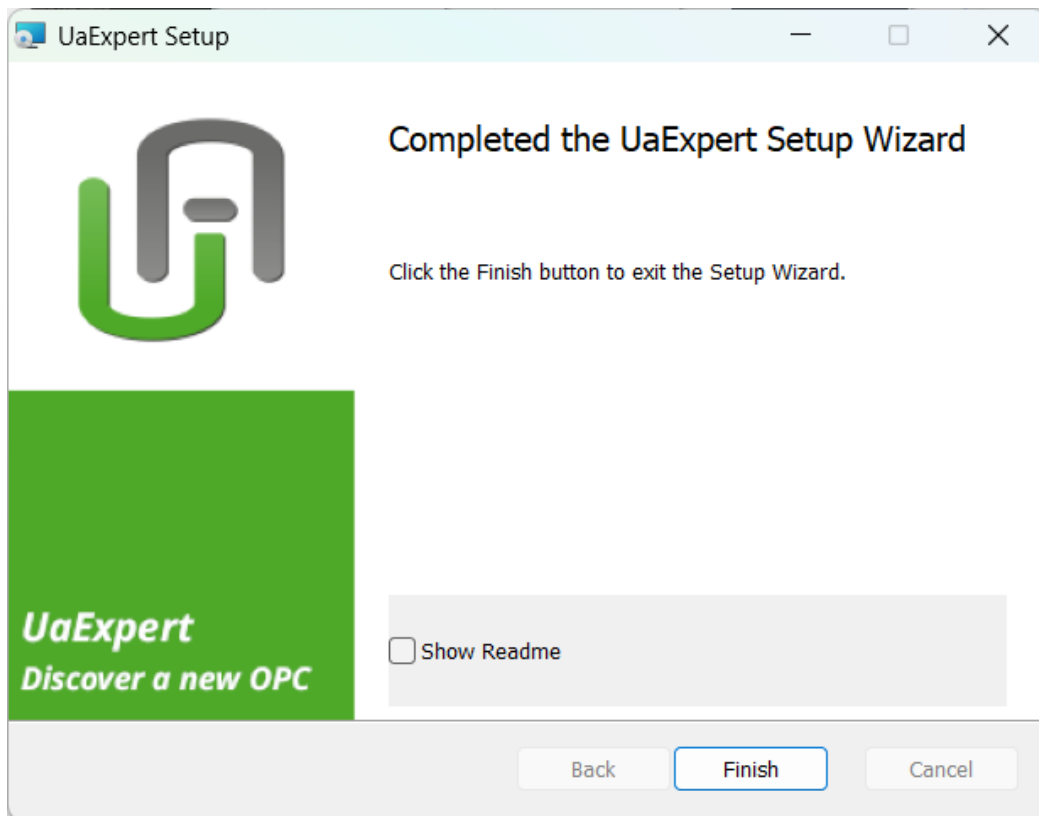


圖 2-2 / 2-3 / 2-4 安裝精靈的三個關鍵頁面。Custom Setup 不用動，預設就會把整包裝進 `C:\Program Files\UnifiedAutomation\`，UaCPPServer 也在裡面。

啟動 UaCPPServer 之後，console 會印出：

```
Server opened endpoints for following URLs:
opc.tcp://<電腦名稱>:48010
```

記下 **port 48010**，但不要抄那個主機名。這支 Client 的 URL 主機部分必須是 **IP**（§2.2.1），填電腦名稱一定連不上。在 PC 上執行 `ipconfig` 查 IP。

手上如果已經有任何一台 OPC UA Server（現場的 PLC、閘道器，或另一台跑 Server 測試工具的 QEC），直接拿來連就好，這一步可以整個跳過——見 §5.2。

2.2 程式架構說明

`QEC_OPCUA_CLIENT_HMI.ino` 約 1,380 行，分成 16 個編號分區，可以分成五層來看：

分層	代表檔案 / 函式	功能
Network Settings	<code>cfgLoad()</code> / <code>cfgSave()</code> / <code>cfgDefaults()</code> / <code>parseIp()</code> / <code>maskValid()</code> / <code>urlProblem()</code> / <code>Ethernet.begin()</code>	EEPROM 讀寫與驗證、靜態 IP 設定、endpoint URL 驗證
HMI	<code>myhmi.h</code> / <code>myhmi.cpp</code> 、 <code>Hmi.*</code> 、 <code>qlog()</code> / <code>setStatus()</code> / <code>setCfgStatus()</code>	五頁版面、觸控數字鍵盤與全鍵盤、面板 log、狀態列

分層	代表檔案 / 函式	功能
Application Logic	<code>handleHmiEvents()</code> / <code>brBrowse()</code> / <code>nodeLoad()</code> / <code>doWrite()</code> / <code>doWatchStart()</code> / <code>favToggle()</code>	按鍵處理、位址空間瀏覽、屬性讀取、依型別寫入、訂閱、最愛
Task / Runtime	<code>setup()</code> / <code>loop()</code> / <code>helperYield()</code> / <code>BEGIN_HMI_EVENT_PROC</code>	單執行緒非阻塞排程：20 ms client iterate、200 ms UI 同步、1 s LIVE 重讀
OPC UA Client	<code>UA_Client_new()</code> / <code>UA_Client_connectAsync()</code> / <code>UA_Client_Service_browse()</code> / <code>UA_Client_Service_read()</code> / <code>UA_Client_MonitoredItems_createDataChange()</code>	連線狀態機、瀏覽、批次讀取、寫入、訂閱

2.2.1 網路設定

設定放在 **EEPROM bank 0 (EEPROM_16K)** 的前 676 bytes：

addr	內容
0	magic 'Q'
1	magic 'C'
2	version (目前 5)
3–6	QEC IP
7–10	Subnet Mask
11–14	Gateway
15–78	目標 server URL (64 bytes)
79–142	最後選取的節點 (64 bytes)
143–526	最愛 NodeId · 6 × 64
527–670	最愛 BrowseName · 6 × 24
671–672	publish 間隔 ms (高、低)
673–674	sampling 間隔 ms (高、低)
675	checksum = XOR(0..674)

開機時 `cfgLoad()` 驗證 magic、version 與 checksum，任一項不符就自動回到 sketch 的 **DEF_*** 預設值，並在 log 標明 `no valid EEPROM config - using defaults`。所以第一次燒錄、或 EEPROM 被清空，機器一樣開得起來。

```
#define DEF_IP      "192.168.1.200"
#define DEF_MASK    "255.255.255.0"
#define DEF_GW      "192.168.1.1"
#define DEF_URL      "opc.tcp://192.168.1.100:48010"
#define DEF_PUBMS    500
#define DEF_SAMPMS   250
```

出廠預設值只在 EEPROM 沒有有效資料時採用。建議把它們設成現場最常用的網段與 server。目標 URL 的主

機部分必須是 IP，不能填電腦名稱。

2.2.2 HMI

所有 log 同時輸出到 **Serial Monitor (115200)** 與 **面板 Log 頁**，時間戳是本地時間。Home、Browse、Node 三頁各有一條狀態列，而 `setStatus()` 是三條一起寫，所以不論你停在哪一頁，看到的都是最後一次動作的結果。

Log 頁寫滿 99 行時會清空，並在第 1 行留下 `----- log full - cleared and restarted -----`，避免「回來只剩幾行、還不知道被清過」。

2.2.3 應用邏輯

週期	做什麼
每一圈	<code>UA_Client_run_iterate(client, 20)</code> —— 推進握手，並承載 Publish 與 DataChange 流量
200 ms	<code>syncUi()</code> ：連線狀態、LED、連線時間、訂閱中的值
1 s	LIVE 重讀，只有停在 Browse 頁且 LIVE 開著時才跑
重試時	遞增退避 1 / 2 / 5 / 10 s，而且只有按過 CONNECT 才會重試
事件觸發	<code>handleHmiEvents()</code> ：換頁、點瀏覽列、READ / WRITE / WATCH / FAV、CONFIRM / DEFAULT / REBOOT

2.2.4 排程與執行

```
void loop() {
  if (client) {
    /* advances the handshake, then carries Publish and DataChange traffic */
    UA_StatusCode rc = UA_Client_run_iterate(client, ITER_MS);
    /* ...LINK_CONNECTING -> LINK_UP, or fail and schedule a retry... */
  }

  /* retry only after CONNECT was pressed, with increasing backoff */
  if (linkState == LINK_IDLE && autoRetry && (long)(millis() - nextTry) >= 0)
    doConnect();

  /* LIVE refresh, only while the Browse page is showing */
  if (liveMode && linkState == LINK_UP && !favMode &&
      Hmi.getCurrentPage() == p2 && millis() - lastLive >= LIVE_MS) {
    lastLive = millis();
    brRefreshPage();
  }
  if (millis() - lastSync >= 200) { lastSync = millis(); syncUi(); }

  BEGIN_HMI_EVENT_PROC
  {
    handleHmiEvents();
  }
  END_HMI_EVENT_PROC
}
```

`BEGIN_HMI_EVENT_PROC / END_HMI_EVENT_PROC` 展開後是一個最多跑 128 次的 `Hmi.eventPop()` 迴圈，結尾呼叫 `helperYield()`——畫面刷新是 `helperYield()` 在做的。這一點在寫任何「等待」邏輯時都要記住：`delay()` 期間面板不會更新，要讓畫面動就得跑 `helperYield()`。

已知限制。 `UA_Client_Service_browse()`、`UA_Client_Service_read()` 與 `UA_Client_writeValueAttribute()` 在 open62541 1.1 裡仍然是同步的，沒有非阻塞版本 可以呼叫。連線本身不會卡住面板，但如果 Read 或 Browse 正好發在斷線的瞬間，最久會阻塞 `SVC_TIMEOUT`。這個值刻意設成 2000 ms：

```
/* ---- Timeouts -----
 * SVC_TIMEOUT    Read / Write / Browse are synchronous in open62541 1.1;
 *                this is how long they can block. Also the async connect timeout.
 * ITER_MS        Maximum time run_iterate() blocks per loop pass.
 * CONNECT_GUARD  Fuse against the state machine sticking in LINK_CONNECTING. */
#define SVC_TIMEOUT 2000
#define ITER_MS     20
#define CONNECT_GUARD 8000UL
```

在區網上 OPC UA 的回應是毫秒等級，2 秒是個寬鬆到幾乎碰不到的上限。

2.2.5 OPC UA Client

連線是一個三狀態的狀態機，在 `loop()` 裡推進：按下 CONNECT 後由 `LINK_IDLE` 進入 `LINK_CONNECTING`，session 被 ACTIVATED 之後轉成 `LINK_UP`；連線失敗會帶著退避回到 `LINK_IDLE`，連線中途斷掉也是一樣。

`doConnect()` 只負責開始，不阻塞任何東西：

```
UA_ClientConfig_setDefault(cc);
cc->timeout = SVC_TIMEOUT;
UA_StatusCode rc = UA_Client_connectAsync(client, cfgUrl);
linkState = LINK_CONNECTING;
```

這支工具開機不會自己連線。測試工具每次面對的 server 都不一樣，自動撥回上一次的位址 沒有實質意義，而且會讓面板在開機後的頭幾秒無法操作。只有按過 CONNECT 才會重試，而且退避是 $1 \rightarrow 2 \rightarrow 5 \rightarrow 10\text{ s}$ ，log 不會被重試訊息塞滿。

2.3 上傳程式

1. 確認 `DEF_IP` 等出廠預設值與現場網段相符
2. 按 IDE 的 **Upload** (→ 箭頭)
3. 編譯 + 上傳約 **30–90 秒**
4. 開啟 Serial Monitor (**115200**)，應該在數秒內看到：

```
=== QEC-M-070T OPC UA Client + HMI ===
QEC_OPCTUA_CLIENT_HMI
QEC_OPCTUA_CLIENT_HMI
config loaded from EEPROM
Ethernet.begin ret=1   QEC IP = 192.168.2.201
ready - check the target URL, then press CONNECT
```

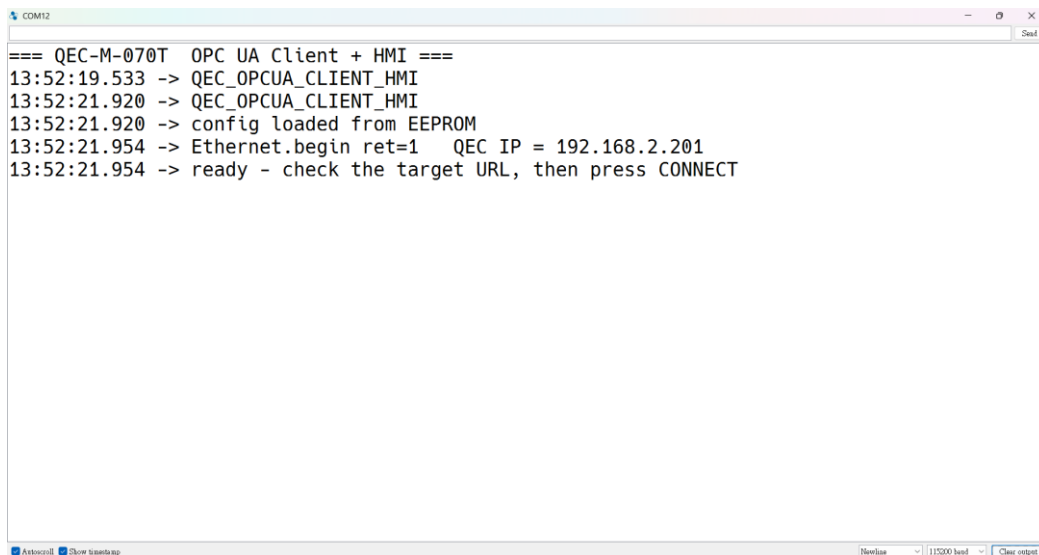


圖2-5 Serial Monitor 的開機訊息。要看的是兩行：`Ethernet.begin ret=1`（是0代表網路沒起來），以及`QEC IP =`顯示設定的位址而不是`0.0.0.0`。sketch 名稱印兩次是正常的：一次是`setup()`裡的`Serial.println()`，一次是`setStatus()`把面板狀態列鏡射到Serial。它不會自動連線——最後一行是提示你按CONNECT。

檢查點	必須是
<code>Ethernet.begin ret=</code>	1（0代表網路沒起來，見 §7）
<code>QEC IP =</code>	你設定的IP，不是 <code>0.0.0.0</code>
最後一行	<code>ready - check the target URL, then press CONNECT</code>

停在這裡是刻意的。現在還沒有連任何東西；先在 Home 頁確認 URL，再按 CONNECT。

3. 連上 Server 與操作節點

3.1 Endpoint

項目	內容
Endpoint 輸入位置	Home 頁的 Target Server 欄位，存入 EEPROM
格式	<code>opc.tcp://<IP>:<port></code> —— 主機部分必須是 IP
SecurityPolicy	None (未編入韌體 <code>UA_ENABLE_ENCRYPTION</code>)
MessageSecurityMode	None
UserTokenType	Anonymous
Discovery (FindServers)	未編入韌體 —— endpoint URL 必須手動輸入
連線	非同步；狀態機在 <code>loop()</code> 裡推進
重試	只有按過 <code>CONNECT</code> 才會；退避 1 / 2 / 5 / 10 s

`urlProblem()` 會在存檔前先驗證這個欄位：

檢查	失敗訊息
以 <code>opc.tcp://</code> 開頭	URL must start with <code>opc.tcp://</code>
長度 64 字以內	URL too long (64 char max)
有主機部分	URL has no host part
主機只含數字與點	URL host must be an IP address, not a hostname

3.2 瀏覽位址空間

瀏覽從 `Objects` (`i=85`) 開始，每一層送出一次 `Browse` service 呼叫：

```
req.nodesToBrowse[0].browseDirection = UA_BROWSEDIRECTION_FORWARD;  
req.nodesToBrowse[0].referenceTypeId = UA_NODEID_NUMERIC(0, UA_NS0ID_HIERARCHICALREFERENCES);  
req.nodesToBrowse[0].includeSubtypes = true;
```

上限	值	意義
<code>BR_MAX</code>	60	單一節點列出的子節點數。超過時狀態列顯示 <code>browse: showing <n> of N child nodes (BR_MAX limit)</code>
<code>BR_ROWS</code>	10	每頁列數；必須與版面一致
<code>BR_PATH</code>	8	最大往下鑽的深度

值直接顯示在清單裡。每次瀏覽之後、以及每次翻頁時，工具會把目前這一頁所有 Variable 的 `ReadValueId` 收

集起來，用一次 Read service 呼叫送出去：

```
rq.nodesToRead      = rvi;  
rq.nodesToReadSize  = (size_t)n; /* ... n <= BR_ROWS ... */  
rq.timestampsToReturn = UA_TIMESTAMPSTORETURN_NEITHER;
```

也就是一頁一次來回，不是一顆節點一次——清單上能直接看到即時值，靠的就是這一點。

3.3 節點屬性

打開一顆節點會讀四個屬性再加上值：

屬性	API
NodeClass	<code>UA_Client_readNodeClassAttribute()</code>
BrowseName	<code>UA_Client_readBrowseNameAttribute()</code>
DataType	<code>UA_Client_readDataTypeAttribute()</code> ，再用 <code>UA_findDataType()</code> 對應成名稱
AccessLevel	<code>UA_Client_readAccessLevelAttribute()</code> → R / W / RW
值 + 兩個時間戳	<code>UA_Client_Service_read()</code> ， <code>UA_TIMESTAMPSTORETURN_BOTH</code>

讀 **DataType** 屬性（而不是從讀回來的值去猜型別）是關鍵：它讓工具可以寫入一顆 從沒讀過的節點，也是 §4.3 輸入框自動切換的依據。

3.4 支援寫入的型別

DataType	輸入元件	鍵盤
Double / Float / Int16 / Int32 / UInt16 / UInt32 / Byte / SByte	NumberInput	數字
Boolean	NumberInput	數字 (0 / 1)
String	TextInput	全鍵盤
其他	NumberInput	數字；按 WRITE 會回 <code>write: <type> not supported by this panel</code>

3.5 訂閱

項目	內容
訂閱數	一次一個；開新的 WATCH 會取代前一個
Publishing interval	<code>cfgPubMs</code> ，50–60000 ms，在 Setup 頁設定
Sampling interval	<code>cfgSampMs</code> ，10–60000 ms，在 Setup 頁設定
要求的時間戳	<code>UA_TIMESTAMPSTORETURN_BOTH</code>

項目	內容
可訂閱的節點	只有 Variable —— 對其他型別按 WATCH 會回 <code>watch: only Variable nodes can be watched</code>

Server 實際核准的修正值會顯示在狀態列：`watching Double   publish=500ms   sampling=250ms`。

4. HMI 頁面功能說明

五頁、Material / Light 主題，版面依 800×480 排。每一頁只做一件事。

4.1 Home — 連線

這一頁只管一件事：連上或斷開。上半是目標 endpoint 與 session 狀態，下半是目前訂閱中的那顆節點。

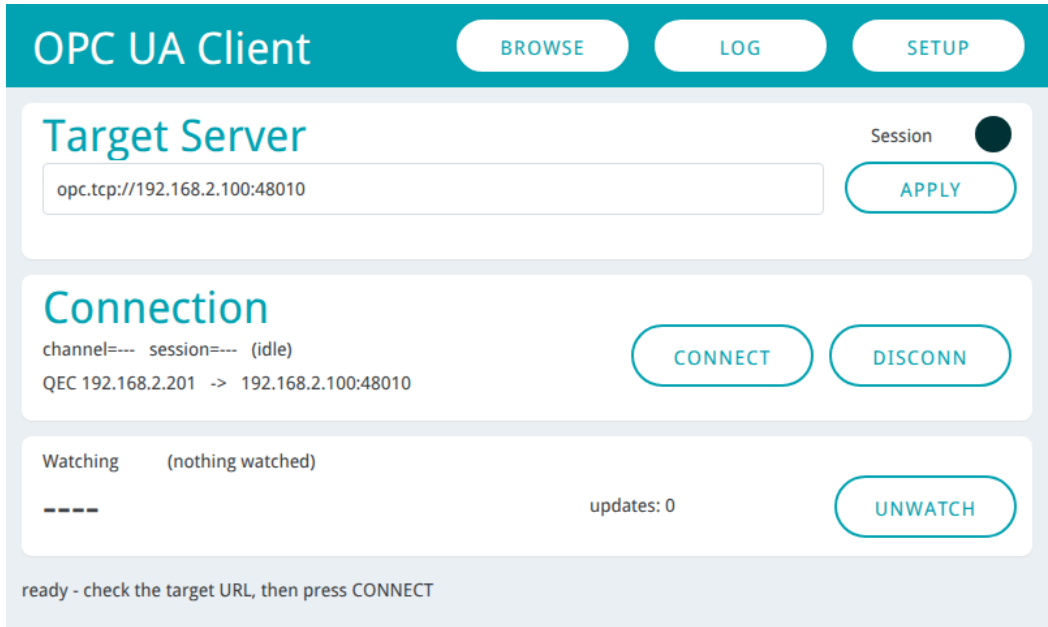


圖 4-1 開機後的 Home 頁：什麼都還沒連，Session LED 微亮，狀態列請你確認 URL 之後按 CONNECT。

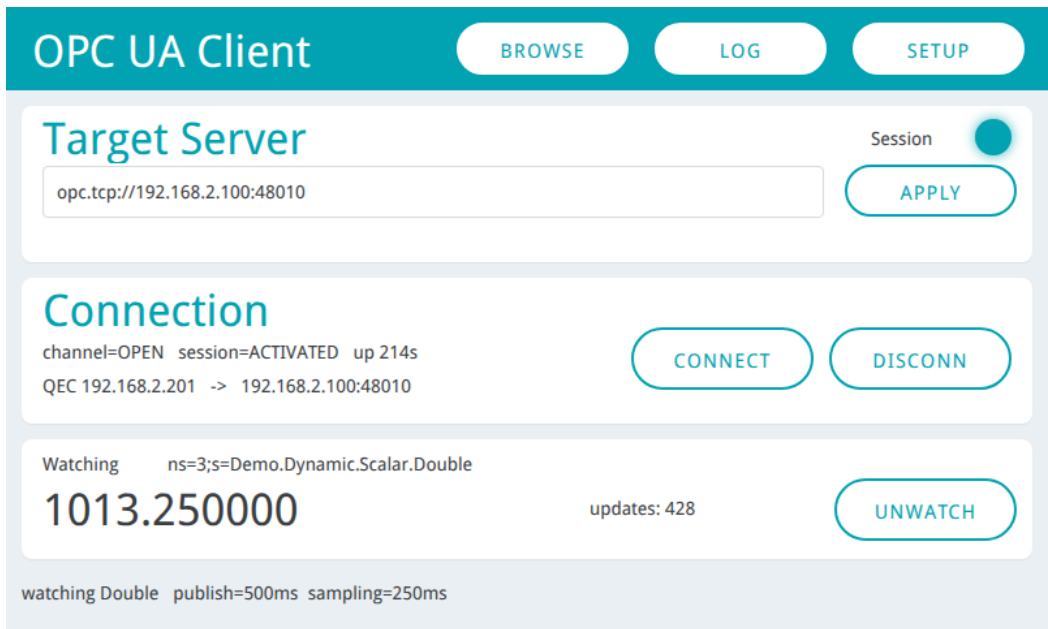


圖 4-2 session 正在跑的 Home 頁：LED 全亮、channel OPEN / session ACTIVATED，並且訂閱著一顆節點。

欄位 / 按鈕	功能
Target Server 欄位	Endpoint。點下去開全鍵盤
APPLY	驗證並存下 URL。如果連線已經在跑，會直接改連新位址

欄位 / 按鈕	功能
Session + LED	session ACTIVATED 時全亮，其他時候微亮
channel= session= up QEC <ip> -> <target>	SecureChannel 與 Session 的原始狀態，加上 session 建立至今的秒數 這台 QEC 的位址，以及目前連線的目標 Server
CONNECT DISCONN	開始連線並啟用自動重試 斷線並停止重試
Watching UNWATCH 狀態列 (底部)	目前訂閱的 NodeId、最新值、DataChange 次數 刪除訂閱 最近一次事件，與 Log 頁最後一行相同
BROWSE / LOG / SETUP	前往 Browse、Log、Setup 頁

事件訊息範例

```
connecting opc.tcp://192.168.2.100:48010
connected
connect FAILED BadDisconnect  retry in 5 s
link lost  retry in 1 s
watching Double  publish=500ms  sampling=250ms
disconnected
```

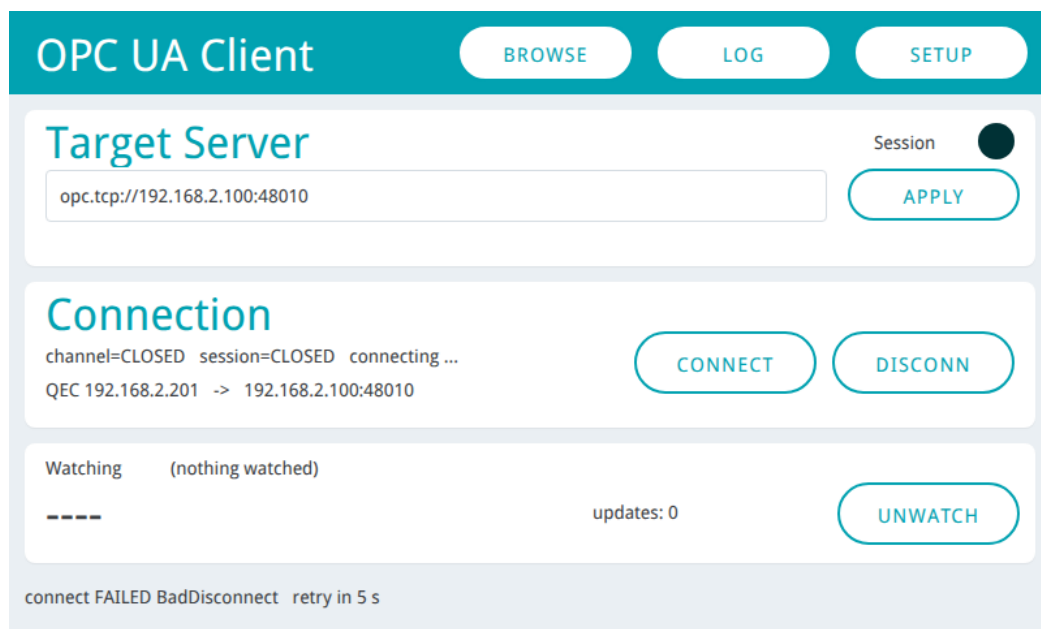


圖 4-3 一次失敗的嘗試：原因與重試倒數共用同一條狀態列，而面板全程完全可以操作。

4.2 Browse — 找節點

一頁十列，往資料夾裡鑽、再走回來。左欄的標記先告訴你每一列點下去會發生什麼，右欄則直接顯示 Variable 的現值。

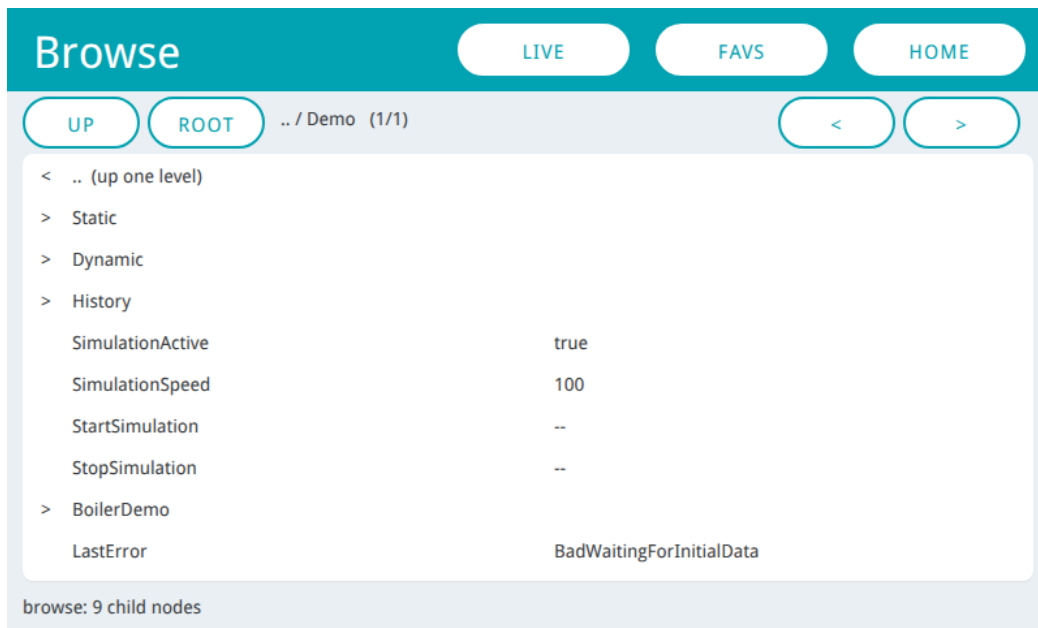


圖 4-4 進到 Demo 底下一層的 Browse 頁。左欄標記每一列的行為；資料夾不顯示值。

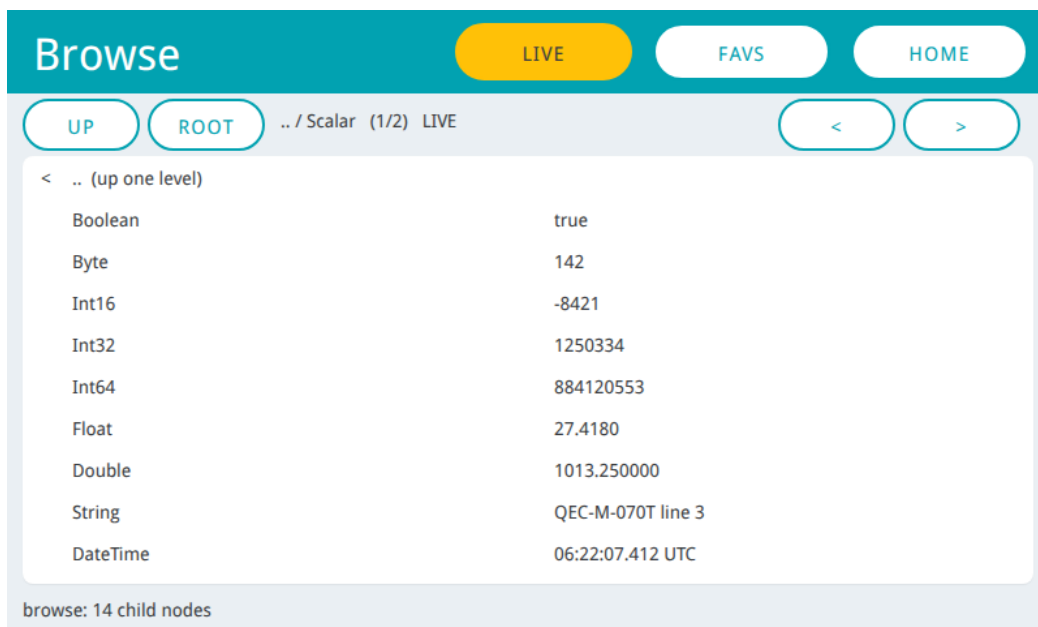


圖 4-5 LIVE 開啟：整頁每秒重讀一次，等於同時監看十類節點。LIVE 按鈕變成琥珀色，麵包屑也帶上 LIVE 標記。

欄位 / 按鈕	功能
左側標記欄	> 這一系列可以往下鑽 (Object) 、< 是 .. 往上一層、空白代表葉節點
名稱欄	BrowseName
值欄	Variable 的現值；讀不到時顯示 StatusCode
麵包屑	最後兩層路徑、頁碼，以及 LIVE 開啟時的 LIVE
UP / ROOT	上一層 / 跳回 Objects
< / >	上一頁 / 下一頁。翻頁會重新讀值
LIVE	切換目前這一頁的每秒重讀。開啟時是琥珀色

欄位 / 按鈕	功能
FAVS	改顯示已存的最愛，而不是瀏覽結果
HOME	回 Home 頁
點列	.. 往上 · Object 往下鑽 · 其他一律開 Node 頁

LIVE 預設是關的。離開 Browse 頁時重讀會暫停，回到這一頁又會繼續。

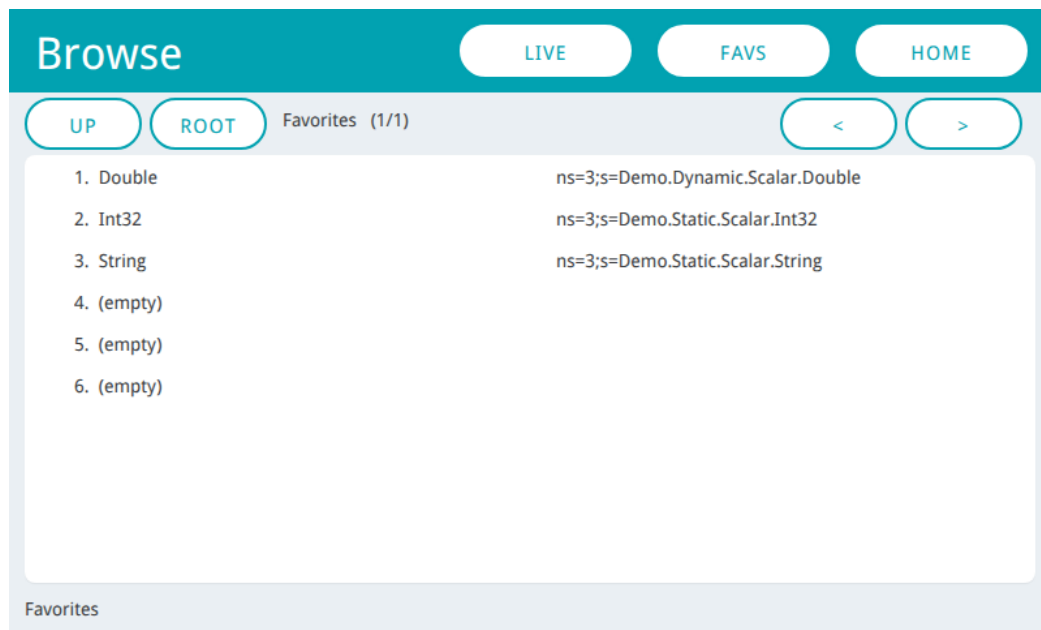


圖 4-6 FAVS 顯示六個儲存格：左邊 BrowseName、右邊完整 NodeId。點一列就在 Node 頁打開它。

4.3 Node — 用節點

一顆節點的屬性、現值與寫入欄位都在這一頁。寫入欄位會跟著節點宣告的 DataType 換成數字框或文字框。

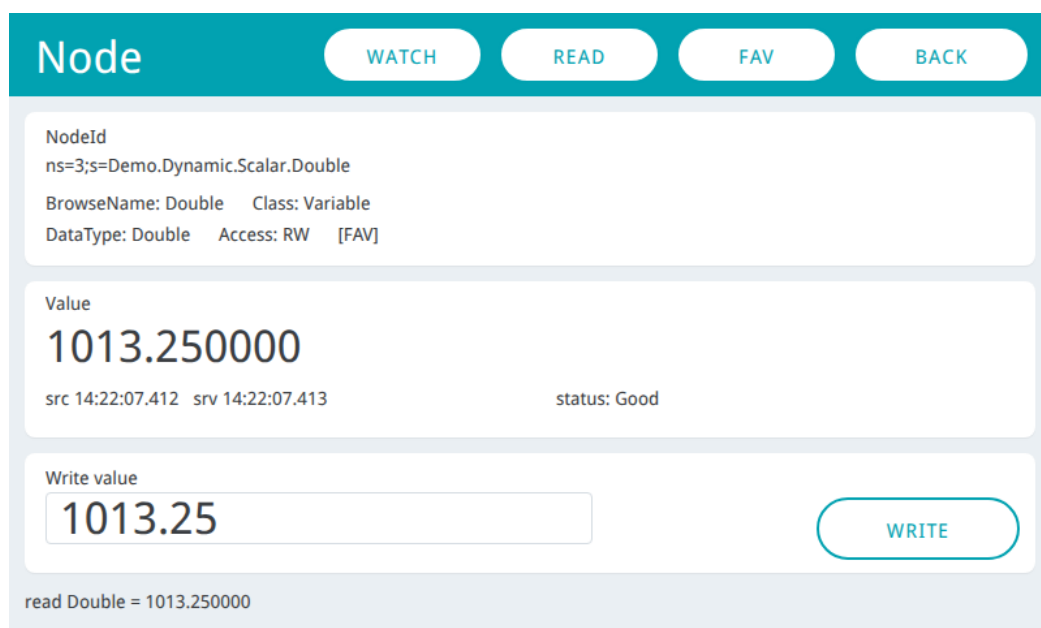


圖 4-7 一顆 Double 節點。寫入欄位是 NumberInput，點下去開數字鍵盤。屬性列尾端的 [FAV] 代表這顆節點已經存進最愛。

Node
WATCH
READ
FAV
BACK

NodeId
ns=3;s=Demo.Static.Scalar.String
BrowseName: String Class: Variable
DataType: String Access: RW

Value
QEC-M-070T line 3
src 14:23:11.008 srv 14:23:11.009 status: Good

Write value

write String -> Good

圖 4-8 一顆 String 節點。同一個位置換成 TextInput，點下去開全鍵盤。

欄位 / 按鈕	功能
NodeId	選中節點的完整 NodeId
BrowseName / Class	來自 BrowseName 與 NodeClass 屬性
DataType / Access	來自 DataType 與 AccessLevel 屬性。節點在最愛裡時會出現 [FAV]
Value	現值，依型別格式化
src / srv	Source 與 Server 時間戳
status	這次讀取的 StatusCode
Write value	輸入元件跟著 DataType 走——數值與布林用數字鍵盤，字串用全鍵盤
WRITE	把欄位寫進節點，然後讀回來
READ	重新讀取所有屬性與值
WATCH	訂閱這顆節點（只有 Variable 可以）
FAV	切換：存進最愛，已存則移除
BACK	回 Browse 頁

圖 4-9 對 AccessLevel 是 R 的節點寫入：Server 回 BadNotWritable，狀態列照實顯示。動手之前 Access 就已經標出來了。

最愛有六個位置。FAV 會把節點存進第一個空位，已存則移除。六個都滿的時候工具會拒絕而不是覆蓋掉舊的：favorites are full (6) - open one from FAVS and press FAV to remove。

4.4 Log — 面板上的即時 Console

機器輸出的每一則訊息都在這裡，與 Serial Monitor 同步。

圖 4-10 Log 頁：開機、連線、瀏覽、讀取、訂閱，以及一次被拒絕的寫入，每一行都有時間戳。

欄位 / 按鈕	功能
TextList (p4t11)	99 行、帶行號、自動捲到最新一行
寫滿時	清空並在第 1 行寫 ----- log full - cleared and restarted -----
CLEAR	清空並從第 1 行重新開始

欄位 / 按鈕	功能
HOME	回 Home 頁

內容與 Serial Monitor 一致 —— 只有開機那兩行 (`=== ... ===` 與 sketch 名稱) 例外，它們在 `Hmi.begin()` 之前就印出去了，只會出現在 Serial。現場只有面板、沒有筆電時，這一頁就是唯一的除錯視窗。

4.5 Setup — QEC 自己的網路與訂閱間隔

這一頁放的是 QEC 自己的網路位址與訂閱間隔 —— 目標 server 的 URL 不在這裡，在 Home 頁。

圖 4-11 Setup 頁，只放 QEC 自己的網路設定；目標 Server 的 URL 在 Home 頁（見 §6.2）。

欄位 / 按鈕	功能
IP / Mask / Gateway	QEC 自己的位址。TextInput，點下去開觸控鍵盤
Active now	開機當下真正生效的值。與上面欄位不同 → 代表改過設定但還沒重開機
Publish (ms)	訂閱的 publishing interval，50–60000
Sampling (ms)	MonitoredItem 的 sampling interval，10–60000
版本 / 目標那一行	燒的是哪一版、存的是哪個 endpoint —— 沒接 Serial 線時很有用
CONFIRM	驗證後寫入 EEPROM
DEFAULT	回到 sketch 的 <code>DEF_*</code> 出廠預設值並寫入。需 3 秒內按兩次
REBOOT	軟體重開機讓網路設定生效。需 3 秒內按兩次
BACK	回 Home 頁

CONFIRM 的驗證規則

任一項不通過就顯示原因、不寫入 EEPROM：

檢查	失敗訊息
IP 為四段 0-255	IP address format error
Subnet 為四段 0-255	Subnet mask format error
Gateway 為四段 0-255	Gateway format error
IP 第一段不得為 0	IP address must not start with 0
遮罩必須連續	Subnet mask must be contiguous, e.g. 255.255.255.0
IP 不能是網路位址	IP is the network address of this subnet
IP 不能是廣播位址	IP is the broadcast address of this subnet
Publish 間隔 50-60000	Publish interval must be 50 - 60000 ms
Sampling 間隔 10-60000	Sampling interval must be 10 - 60000 ms

圖 4-12 Subnet 填 255.0.255.0 之後按 CONFIRM，狀態列說明被拒絕的原因，值不會被存下來。

遮罩連續性用位元判斷（255.0.255.0 這種會被擋掉，存進去重開機就整台連不上）；網路 / 廣播位址是依遮罩判斷的，所以 192.168.5.0 在 /16 底下仍然合法。

Gateway 不在同網段只提示不阻擋（內網測試常常不需要 gateway），存檔訊息會變成 Saved: ... - WARNING gateway not in this subnet。

二次確認

DEFAULT 與 REBOOT 都會覆寫現場設定或中斷連線，所以都要在 3 秒內按兩次：

第一次按 → 狀態列：Press REBOOT again within 3 s to restart the device
第二次按 → 執行

兩顆鈕互斥：按 **REBOOT** 會取消已經按過一次的 **DEFAULT**，反過來也一樣；按 **CONFIRM** 則是兩個都取消。進入 Setup 頁時也會全部歸零。

5. 實際跑一次：連上第一台 Server

5.1 這一章要做什麼

找一台 OPC UA Server 來連，走一次最短的路：連上 → 瀏覽節點 → 讀一次、寫一次 → 訂閱一顆會變的節點。約十分鐘，中間不需要寫任何程式，也不需要 PC 端的 OPC UA Client。

跑完這一遍，你就確認了三件事：QEC 連得到對方、面板上看得到對方有哪些節點、而且讀寫與訂閱都成立。接下來要接的是 demo server、現場的 PLC 還是另一台 QEC，對這支工具來說沒有差別——它面對的都是同一套標準介面。

5.2 開始之前

1. QEC MDevice 已燒錄 `QEC_OPCUA_CLIENT_HMI`，Serial 顯示 `ready - check the target URL, then press CONNECT`
2. 網路線插在 **Gigabit RJ45** 孔（不是那兩個 10/100 的 EtherCAT 孔）
3. 有一台 OPC UA Server 在跑而且連得到——任何一台都可以，見下方說明。本章用的是 Unified Automation 的 demo server（安裝步驟見 §2.1.3）；記下它的 IP 與 port（預設 48010）。它的 console 印出來的 endpoint 用的是 PC 的電腦名稱——IP 要從 `ipconfig` 查，不要抄那一行
4. PC 與 QEC 在同一個網段（在 PC 上用 `ipconfig` 確認）
5. Server 接受 **Security None / Anonymous**——這支 client 未編入韌體加密

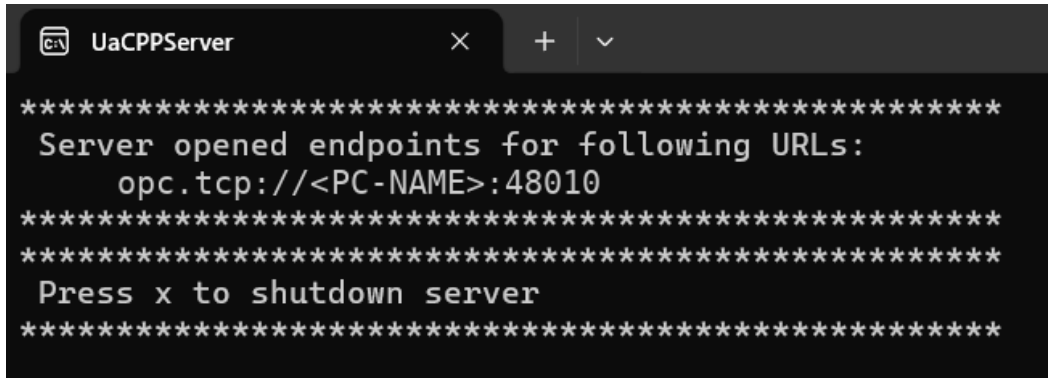
任何一台 OPC UA Server 都能用

這支工具沒有任何地方是為 demo server 特製的。網路上如果已經有 server，直接把 QEC 指過去就好，完全不必安裝——連線、瀏覽、讀取在每一種情況下都一樣，差別只在圖上的節點名稱：

要連的 Server	要輸入的 endpoint	補充說明
第二台 QEC 跑 <code>QEC_OPCUA_Server_HMI</code>	<code>opc.tcp://<那台 QEC 的 IP>:4840</code>	最單純的一組：一台當 Server、另一台瀏覽它，兩邊各有自己的 log 可以對照。它的十顆 <code>QEC.*</code> 節點在 Server 手冊裡有完整說明，而 <code>QEC.Counter</code> 每秒 +1，正是 LIVE 與 WATCH 需要的東西
現場既有的 PLC、閘道器或 SCADA server	它自己公布的位址	先確認它允許 Security None / Anonymous ，以及它的節點可以安全地被寫入。對正在運轉的設備，先 READ 再 WRITE
Unified Automation 的 demo server	<code>opc.tcp://<PC 的 IP>:48010</code>	本章接下來用的就是它。它的 <code>ns=3;s=Demo.*</code> 樹涵蓋了面板寫得到的所有型別

兩台 QEC，各一條線。一台跑 Server sketch 在 `192.168.2.200:4840`、這一台在 `192.168.2.201`，輸入 `opc.tcp://192.168.2.200:4840` 按 **CONNECT**。瀏覽到 **Objects** ▶ **QEC** 就會看到 Server 的十顆節點與即時值；**Setpoint** 與 **Analog** 依設計是唯讀的，所以它們也剛好是 §5.5 第 15 項 **BadNotWritable** 的現成測試對象。

唯一的硬性要求是 endpoint 的主機部分。不管挑哪一台 server，主機部分都必須是 IP 位址（§2.2.1）。



```
UaCPPServer
*****
Server opened endpoints for following URLs:
opc.tcp://<PC-NAME>:48010
*****
Press x to shutdown server
*****
```

圖5-1 PC 上執行中的UaCPPServer。Server opened endpoints for following URLs 代表它正在監聽，而那個URL 尾端的port——48010——就是要填進QEC 目標URL 的數字。它印出來的主機部分是PC 的電腦名稱，而那正是你不能填進QEC 的東西（見§2.2.1）：在PC 上執行ipconfig 改用IP，也就是opc.tcp://192.168.2.100:48010。本圖中電腦名稱已遮蔽。

5.3 步驟

#	步驟	動作
1	確認 QEC 自己的位址	SETUP → 確認 Active now 與你的網段相符 → BACK
2	輸入 endpoint	在 Home 點 Target Server 欄位，輸入 opc.tcp://<server IP>:48010，確認
3	存起來	按 APPLY —— 狀態列顯示 target saved: ...
4	連線	按 CONNECT
5	確認 session	Session LED 亮起，那一行顯示 channel=OPEN session=ACTIVATED，狀態列顯示 connected
6	瀏覽	按 BROWSE。清單已經是 Objects 了 —— 連上時工具會自動瀏覽一次
7	往下鑽	點有 > 標記的列往下；點 .. 或 UP 回來
8	同時監看十顆節點	找一個裝滿 Variable 的資料夾按 LIVE —— 按鈕變琥珀色，整頁每秒重新讀一次
9	打開一顆節點	點沒有標記的列，Node 頁會帶著屬性與值打開
10	存起來	按 FAV；屬性列出現 [FAV]

5.4 連上之後

Home 頁顯示 channel=OPEN session=ACTIVATED、LED 全亮，Log 頁記錄了 connecting ... 然後 connected。

按 **BROWSE** · **Objects** 底下的子節點已經列出來，而且 **Variable** 的值也已經填好了——一顆節點都不必打開。

那一整頁的值是一次批次 **Read** 抓回來的（一頁一次來回，不是一顆一次），這正是面板上的瀏覽清單能真正拿來用的原因。

5.5 動手做四件事

在 **Browse** 頁點一顆沒有標記的列，**Node** 頁就會帶著它的屬性打開。接著做四件事：

1. **讀一次** —— 按 **READ**，值與兩個時間戳更新，狀態列顯示 **read <name> = <value>**。屬性列上的 **DataType** 與 **Access** 是接下來兩步的依據。
2. **寫一個數值** —— 在寫入欄位輸入 **1.2345**、按 **WRITE**，狀態列回 **write <name> -> Good**，值卡片變成 **1.234500**。
3. **寫一個字串** —— 換一顆 **String** 節點打開。同一個位置的寫入欄位自動變成文字框、而且已經預填現值，點下去開的是全鍵盤而不是數字鍵盤。改一個字按 **WRITE** 就好。輸入框是依節點宣告的 **DataType** 切換的，不是靠猜的。
4. **訂閱一顆會變的節點** —— 按 **WATCH**，然後 **BACK** → **HOME**。Home 頁的 **Watching** 區塊會顯示 **NodeId**、一直在變的值，以及不斷上升的 **updates:** 計數。狀態列那行 **publish=<n>ms sampling=<n>ms** 是 **Server** 實際核准的間隔，常常不等於你要求的值。

過程中會一併看到的現象

你會看到	意思
進到裝滿 Variable 的資料夾，右欄的值立刻就有	一次批次 Read 把整頁抓回來（§3.2）
翻頁之後值也已經讀好了	換頁會重新批次讀一次
按 LIVE 之後按鈕變琥珀色、麵包屑出現 LIVE	每秒重讀整頁；離開 Browse 頁會暫停，回來繼續
對 Access 是 R 的節點按 WRITE 會回 BadNotWritable	存取權保護生效，這是預期行為
輸入 opc.tcp://MYPC:48010 被擋下來	主機部分必須是 IP（§2.2.1），存檔前就先擋掉
拔掉網路線後出現 link lost retry in 1 s	斷線會自動重連，退避 1 / 2 / 5 / 10 s
斷電重開後目標 URL 、最愛、間隔都還在	全部存在 EEPROM 裡（§6）

到這裡，**QEC** 當 **OPC UA Client** 的能力就算確認可用了。節點怎麼瀏覽、屬性怎麼讀、哪些型別寫得進去，都整理在 §3；接下來可以做什麼在 §9。

5.6 這些訊息是正常的

第一次用這支工具的人最常把下面幾則當成故障，其實都不是：

現象	說明
立刻 <code>connect FAILED BadDisconnect</code>	IP 或 port 錯、server 沒在跑，或防火牆。先從 server 那一端確認 endpoint
URL host must be an IP address, not a hostname	這是正常的 —— 要填 server 的 IP，不是它的電腦名稱（見 §2.2.1）
連上了，但 <code>BROWSE</code> 是空的	Server 在 <code>Objects</code> 底下沒有東西，或這個 session 沒有瀏覽權限。用 PC 端的 client 連同一個 endpoint 對照
本來以為會展開的一列卻開了 Node 頁	這一版的預期行為 —— 只有 NodeClass 是 <code>Object</code> 的節點會往下鑽。見 §7
某些列的值欄是空的	那些列不是 Variable（資料夾、Method），本來就沒有值可讀
值欄顯示的是 StatusCode	節點存在但讀不到 —— 顯示的就是 server 的回答

6. 設定的持久化

要讓工具回到你離開時的樣子，需要的東西全部放在 EEPROM：QEC 自己的網路、目標 endpoint、最愛，以及訂閱間隔。

項目	目的	作法
EEPROM 設定區	讓 QEC 網路、目標 endpoint、最愛與訂閱間隔在斷電後保留	bank 0 前 676 bytes，magic + version + XOR checksum；任一項不符自動回到 DEF_*
Active now 那一行	分清「已存但還沒生效」與「正在跑」	Setup 頁，顯示開機當下生效的那組值；每次進 Setup 頁由 activeToUi() 重新填入
REBOOT 鈕	讓「存檔 → 生效」在面板上就能走完	3 秒內按兩次

6.1 什麼要重開機、什麼不用

設定	何時生效
QEC 的 IP / 遮罩 / 閘道	重開機時 —— Ethernet.begin() 只在 setup() 裡跑一次
目標 server URL	下一次 CONNECT —— 按 APPLY，若連線已在跑會立刻改連
Publish / Sampling 間隔	下一次 WATCH
最愛	立即

只有 QEC 自己的網路需要重開機，所以 Setup 頁上只有那一區標了 (restart required)。

6.2 為什麼目標 URL 放在 Home 頁

把工具指向另一台 server 是最常做的動作，所以這個欄位就放在管理連線的那一頁，而不是埋進設定畫面裡。APPLY 會驗證、存檔，而且——如果 session 已經在跑——一步就改連新位址。

6.3 EEPROM 版本變更

這個佈局帶了一個版本位元組 (EE_VER，目前是 5)。當韌體更新改了佈局，版本就對不上，cfgLoad() 會刻意退回預設值，而不是把舊的位元組當成新的來讀。

改過 EE_VER 的韌體更新之後，第一次開機會顯示 no valid EEPROM config - using defaults。IP 與目標 URL 要再輸入一次。這是刻意的設計——誤讀舊佈局的後果更糟。

7. 疑難排解

問題	可能原因	解法
編譯錯誤 'p3nin1' was not declared in this scope	改過 data_project.hpj 的版面之後沒有用 86HMI 設計工具 存檔 —— myhmi.h/cpp 還是舊的，缺少新元件	開設計工具載入 data_project.hpj 存檔一次，重新產生 myhmi.h/cpp ，再編譯。沒有改版面的話不會遇到（這兩個檔隨專案附上）
Serial 印 QEC IP = 0.0.0.0、Ethernet.begin ret=0	網路線插錯孔 / IP 與他人衝突 / EEPROM 值異常	確認插 Gigabit 孔；換一個沒被占用的 IP；面板 SETUP → DEFAULT (×2) → REBOOT (×2)
URL must start with opc.tcp://	少了 scheme	完整輸入含 opc.tcp:// 的 endpoint
URL host must be an IP address, not a hostname	填了電腦名稱	這是刻意的 —— DJGPP 沒有 getaddrinfo() 。改用 server 的 IP
connect FAILED BadDisconnect 一直重試	Server 沒在跑、port 錯、防火牆，或網段不同	先用 PC 端 client 確認 endpoint；在 Setup 頁把 Active now 與你的網段對一次
連上了，幾秒後 link lost	Server 關掉了這個 session (安全性或 user token 政策)	這支 client 只提供 Security None / Anonymous 。請在 server 上開一個 anonymous 的 None endpoint
在 PC client 上會展開的節點，在這裡卻開了 Node 頁	只有 NodeClass 是 Object 的節點會往下鑽。帶子節點的 Variable (例如 DataAccess AnalogItem) 會當成節點打開，而不是展開	這一版的已知限制。那些子節點在 PC client 上仍然拿得到；在這裡讀寫這類節點本身也一切正常
browse: showing <n> of N child nodes (BR_MAX limit)	這類節點的子節點超過 BR_MAX	在 .ino 裡把 BR_MAX 改大後重新燒錄，或改從子節點比較少的資料夾進入
某一列的值欄是空的	那一列不是 Variable	預期行為 —— 只有 Variable 會被讀
write: data type unknown - press READ first	DataType 屬性讀不到	按 READ ；還是失敗的話代表該 server 沒有公開這個屬性
write: <type> not supported by this panel	節點不是支援的純量型別	見 §3.4
watch: only Variable nodes can be watched	對 Object 或 Method 按了 WATCH	預期行為
favorites are full (6)	六個位置都滿了	FAVS → 打開一顆 → FAV 移除
斷線時面板停頓一兩秒	預期行為 —— open62541 1.1 的 Read / Write / Browse 是同步的	最壞情況是 SVC_TIMEOUT (2000 ms)。連線本身永遠不會阻塞
REBOOT 按了沒反應	這個 IDE 版本的 port IO 行為不同	改成手動斷電重開就好；其他功能都不受影響
每次開機都顯示 no valid EEPROM config	佈局版本改過 / EEPROM 沒寫入成功或 checksum 不符	在 Setup 頁按一次 CONFIRM ；仍失敗則 EEPROM 可能異常

問題	可能原因	解法
狀態列訊息被截字	那個 Label 的寬度被設計工具依文字縮小了	見 §8 的警告；確認狀態列 Label 寬度仍在 700 px 以上

8. 完整程式與下載

QEC_OPCUA_CLIENT_HMI\ ├─ QEC_OPCUA_CLIENT_HMI.ino ├─ CHANGELOG.md ├─ myhmi.h ├─ myhmi.cpp └─ data\ └─ _project.hpj	主程式（約 1,380 行、16 個分區，手改這個） 程式的設計決策與實測記錄 86HMI 設計工具產生（勿手改） 86HMI 設計工具產生（勿手改） HMI 版面專案（5 頁，用設計工具改）
--	--

檔案	說明
QEC_OPCUA_CLIENT_HMI.ino	全部的應用邏輯，分成 16 個編號分區，檔案最上方有索引。五層結構見 §2.2
data_project.hpj	HMI 版面。改版面請用設計工具，不要直接編輯 myhmi.*
CHANGELOG.md	含已知問題與實作注意事項：連線為什麼非改成非同步不可、browse 的指標別名 bug 與修法、Node 頁為什麼要疊兩個輸入元件

⚠ 要改 HMI 版面的話先讀這段：Label 寬度會被設計工具覆寫。在 .hpj 的 JSON 裡設 Label 的 width 沒有用——用 86HMI 設計工具存檔時，工具會依「設計期文字」重新 auto-fit 每個 Label 的寬度（實測約 8.15 px/字）。所以本專案的佔位文字都刻意填成「最長的現實情況」：狀態列放 94 個字元的佔位字串，十個瀏覽列放 72 個字元的。改動這些 Label 的文字之後，請回頭確認寬度沒有被縮小，否則執行時的長訊息會被截字。

Node 頁刻意放了兩個重疊的輸入元件（p3nin1 與 p3tin1）。設計工具的預覽會看到它們疊在一起，那是正確的。不要把它們分開——sketch 會把不適用的那個隱藏掉。

改完版面記得存檔，myhmi.h / myhmi.cpp 才會跟著更新。

範例程式 ZIP 下載：https://www.qec.tw/wp-content/uploads/2026/09/QEC_OPCUA_CLIENT.zip

9. 後續延伸

確認 QEC 能當 OPC UA Client 之後，接下來就是把它接進你自己的系統。

9.1 對上現場既有的 Server

這一步通常不用改 QEC 這邊的任何東西。現場的 PLC、閘道器、上位系統只要是標準 OPC UA Server，在 Home 頁填入它的 endpoint、按 **APPLY** → **CONNECT** 就會連上。

要事先確認的只有三件事：

項目	要求
Endpoint	<code>opc.tcp://<對方的 IP>:<port></code> —— 主機部分必須是 IP，不能填電腦名稱 (§2.2.1)
安全性	對方要開放 SecurityPolicy None 、 Anonymous ；這一版未編入韌體加密
節點	由對方定義。用面板直接 Browse 就看得到的，不需要事先拿到節點清單

對正在運轉的設備，先 **READ** 看清楚 **Access** 與 **DataType** 再決定要不要 **WRITE**。

9.2 兩台 QEC 對接

一台跑 Server 工具、一台跑這支 Client 工具，中間不放 PC。這是把一條連線從頭到尾 走完最快的方式，而且兩邊都有自己的 Log 頁可以對照，出問題時很容易判斷是哪一端。

Server 那一端的十顆 QEC.* 節點見 [《QEC OPC UA Server 測試工具 — 使用手冊》](#)；其中 QEC.Counter 每秒 +1，最適合拿來看 **LIVE** 與 **WATCH**。

9.3 QEC 反過來當 Server

如果要做的是「QEC 公開自己的資料、由 SCADA 連進來」，那是另一個方向，見上面那份 Server 手冊。兩種角色可以在同一支 sketch 裡並存，library 兩邊都支援。

9.4 可以再延伸的功能

方向	內容
瀏覽清單加篩選	面對每個資料夾有上百個子節點的 server，翻頁很慢。加一個篩選欄位是最明顯的下一步，代價是十列裡少一列
讓帶子節點的 Variable 也能展開	把往下鑽的規則從 Object 延伸出去，讓 DataAccess 結構像在 PC client 上一樣展開
支援更多寫入型別	<code>doWrite()</code> 目前涵蓋純量內建型別。陣列、 LocalizedText 、 ByteString 各自需要自己的輸入處理

方向	內容
加入 EtherCAT	非阻塞的 <code>loop()</code> 已經留好空間：在每次 <code>UA_Client_run_iterate()</code> 之後更新 EtherCAT，把訂閱到的節點值轉成目標位置

9.5 上線之前要處理的安全性

這一版未編入 `UA_ENABLE_ENCRYPTION`，只連得上 SecurityPolicy None、驗證 Anonymous 的 Server —— 連線內容在網路上是明文的。用於測試與封閉的產線網段沒有問題，但要跨到辦公網路或對外之前，請先用防火牆或獨立網段把它隔開。

10. 聯絡我們

業務諮詢	info@qec.tw
技術支援	info@icop.com.tw
官方網站	https://www.qec.tw

QEC OPC UA Client Test Tool - User Manual v 1.0 ICOP Technology Inc.